# GD2S02 – Software Engineering for Games

## Looking over Software Engineering Practices

# Overview

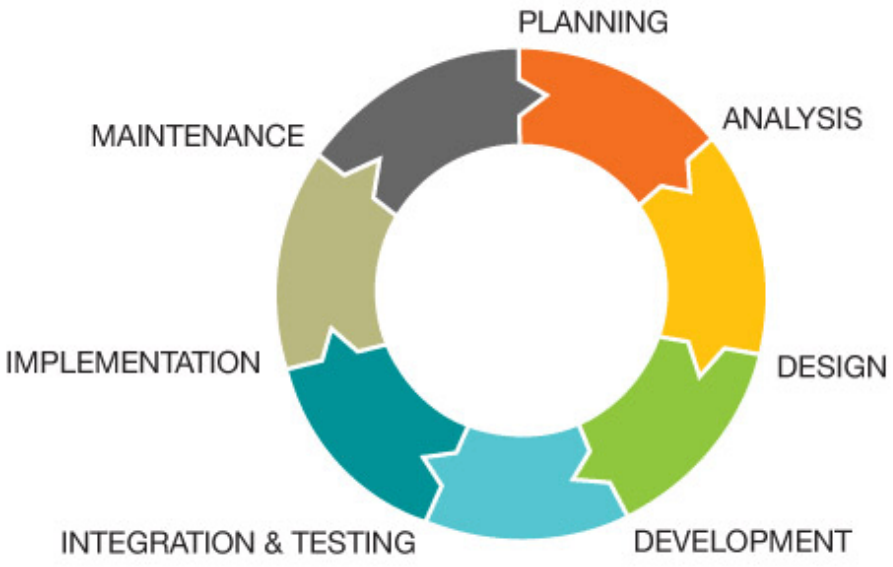
- SDLC
  - Phases of SDLC
- Iterative Process Models
  - Prototyping
  - Spiral
  - Game Development with Spiral Approach
  - Agile Practices
  - High Concept Document

# Main Goal of Software Development Life Cycle

- Manage the process well!
- A well-managed process will produce high quality products on time and within budget.
- Best indicator of how well a software process has worked is the quality of the deliverables produced.

# Main Phases of Software Development Life Cycle

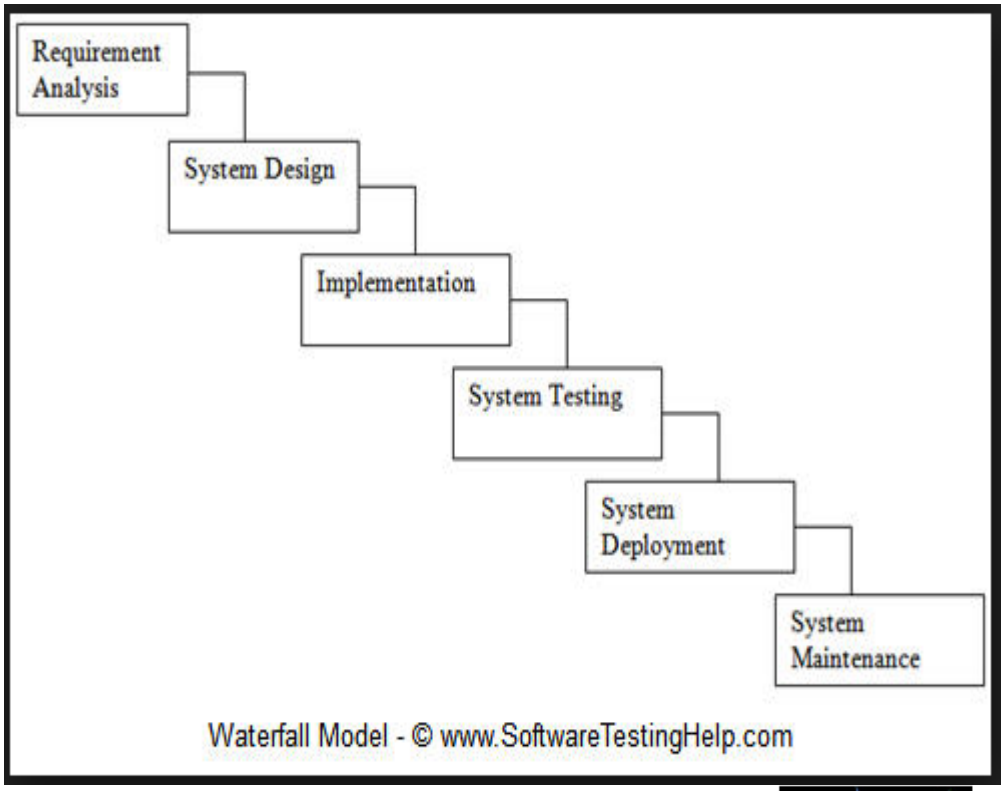
- All SDLC contains the following phases:
  - Requirements Gathering and Analysis
  - Design
  - Implementation (Development)
  - Testing
  - Maintenance




http://www.webserv.ca/developmentCycle.php

# Linear Process Models

- Each phase in the SDLC must be completed before the next phase can start.
- Phases do not overlap
- Example is the Waterfall Model

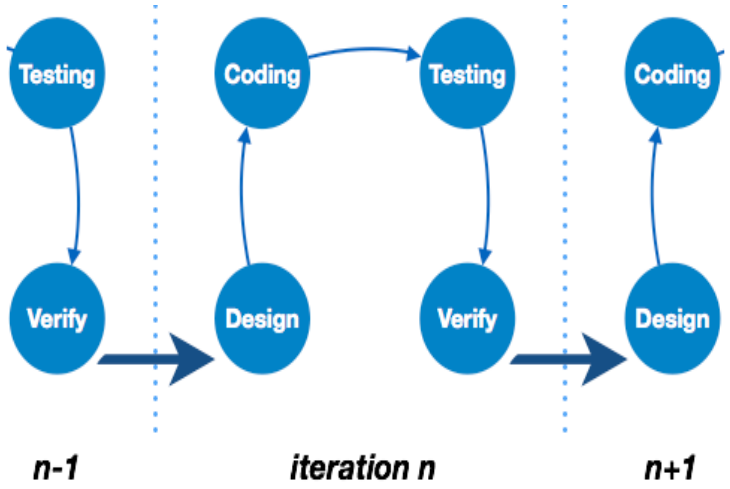


# Linear Process Models

- The main advantages of the waterfall model are:
  - Simple and easy to understand and use.
  - Easy to manage.
  - Works well with small projects, where the requirements are well understood.
- The main disadvantages of the waterfall model are :
  - No working product is available until late of the SDLC.
  - It can not accommodate changing requirements.
  - No scope adjustment is allowed.
  - Not suitable for projects where requirements are not well understood or at high risk of changing.

# Iterative Process Models (Incremental)

- Starts with an implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented.
- This process is repeated producing a new version of the software at the end of each iteration.
- Feature code is designed, implemented, and tested in repeated cycles.



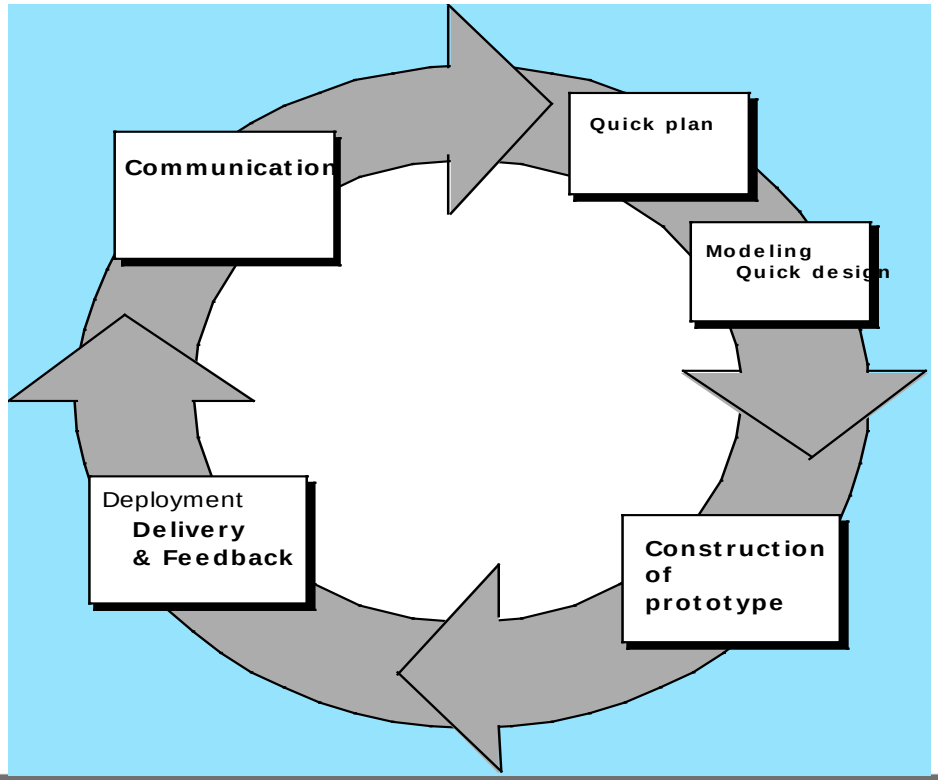

www.tutorialpoint.com

# Iterative Process Models

- Advantages of iterative SDLC Models:
  - Constantly watching over risks
    - Analyse, identify, resolve
  - A working product is available at early phases of the SDLC
  - Less cost to change the scope/functionality
  - Best suited for big projects, where the major requirements can be defined, but some details can evolve with time.

# Prototyping

- Prototyping paradigm assists the Software engineers and the customer to better understand what is to be built **when requirements are fuzzy**.



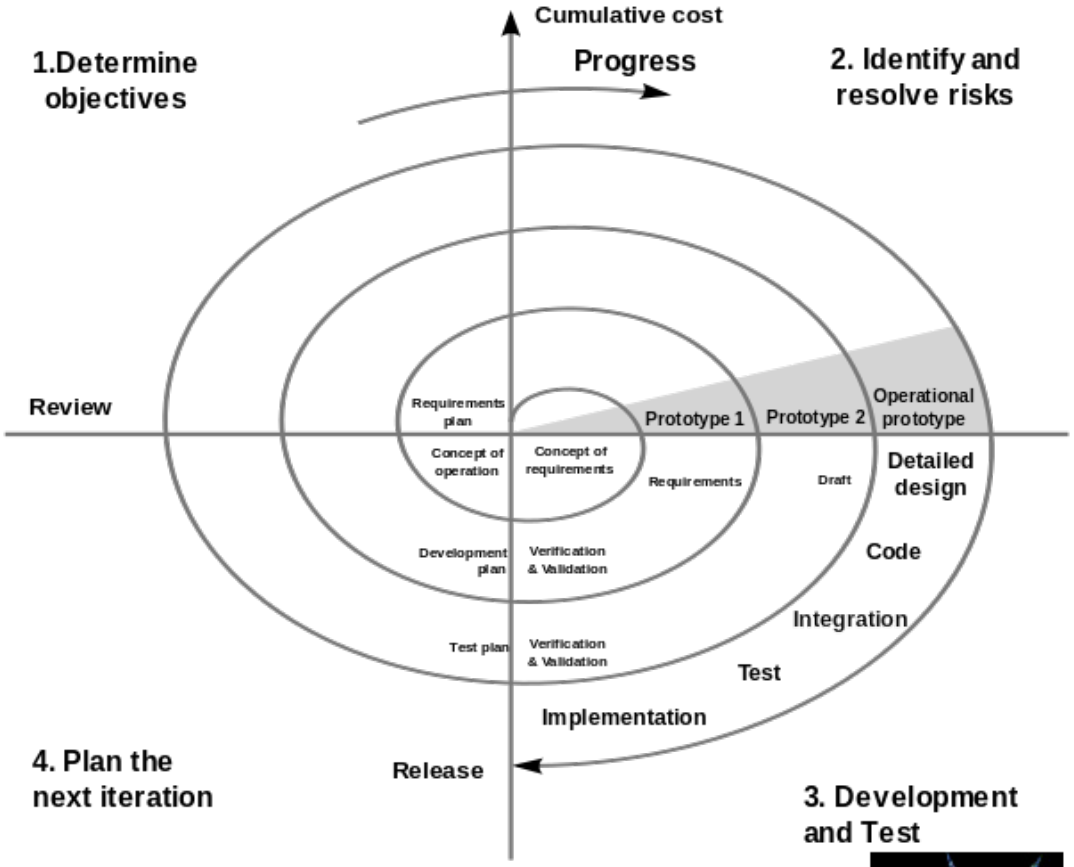

- The prototype is built to serve as a mechanism for defining requirements.



Software Engineering, 9th Edition, by Ian Sommerville, Publisher: Addison-Wesley, Copyright: 2011, ISBN-10: 0137035152

# Spiral Model

- As defined by Boehm, 1988
- It is similar to the incremental model, with more emphasis placed on risk analysis.

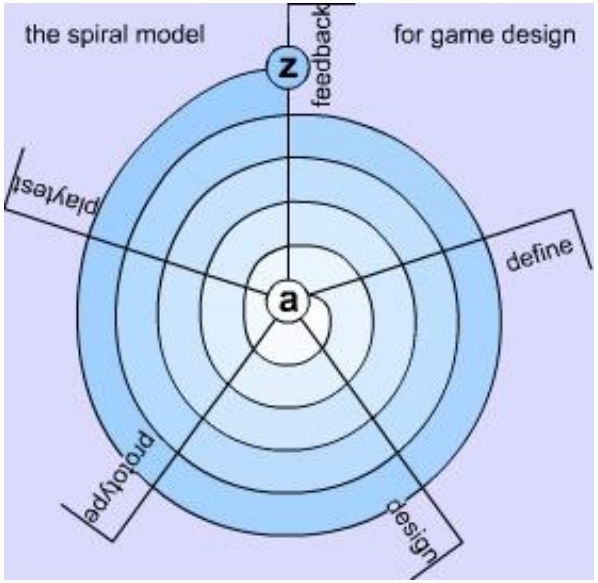


# Spiral Model : Steps

- The new system requirements are defined in as much detail as possible.
- A **preliminary design** is created for the new system.
- A **first prototype** of the new system is constructed from the preliminary design. (a scaled-down system representing an approximation of the characteristics of the final product)
- Create the next prototype based on the findings from the first; **four steps approach.**
- Risk analysis: the entire project can be aborted if the risk is deemed too great. (Risk factors: development cost overruns, operating-cost miscalculation etc.)
- The existing prototype is evaluated, earlier steps are repeated, prototype is refined.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures.

# Game Development with spiral approach

- Define - define the new idea, make an outline, and hash out the basics of your design.
- Design - get down to work writing the rules for the core mechanic, make the basics workable.
- Prototype - build a working model, start to get the look and feel of the physical features of the game.
- Playtest - you can do a solo playtest, an internal playtest, or a blind playtest.




- Feedback - gather, collate, and synthesize your feedback.
- Redefine - go back to the drawing board with what you learned and change your outline...



Ref: http://en.wikiversity.org/wiki/Game_Creation_Guidelines/Spiral_Model_of_Game_Development

# Agile Practices

## Agile Manifesto:

"We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more."

[http://agilemanifesto.org/]

# Agile Practices

- The Agile development model is a type of incremental model.
- Software is developed in incremental, rapid cycles.
- Agile is based on adaptive software development methods, where the team adapts to the changing product requirements dynamically.

# Agile Principles

- The twelve principles of Agile development are:

**Principle 1: Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**

**Principle 2: Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.**

**Principle 3: Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.**

# Agile Principles

- The twelve principles of Agile development are:

**Principle 4: Business people and developers must work together daily throughout the project.**

**Principle 5: Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.**

**Principle 6: The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**

# Agile Principles

- The twelve principles of Agile development are:

**Principle 7: Working software is the primary measure of progress.**

**Principle 8: Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.**

**Principle 9: Continuous attention to technical excellence and good design enhances agility.**

# Agile Principles

- The twelve principles of Agile development are:

**Principle 10: Simplicity -- the art of maximizing the amount of work not done -- is essential.**

**Principle 11: The best architectures, requirements, and designs emerge from self-organizing teams.**

**Principle 12: At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.**

# Agile Practices in Game Design

- These postulates have a strong connection with the context of game development:
  - Individuals and interactions over processes and tools, for the teams are mostly interdisciplinary and creativity plays a great role in game development.
  - Working software over comprehensive documentation, for the game must be played to be evaluated, and documentation has little significance to the end-user.
  - Customer collaboration over contract negotiation, for the subjective character of games makes it hard to define the final outcome. The constant satisfaction of the stakeholders guides the project.
  - Responding to change over following a plan, for it is difficult to predict how certain features will interact in the final result.

# Scrum framework

- **Scrum** is an iterative and incremental agile software development framework for managing product development.
- Scrum uses **Sprints,** sprints are fixed-length iterations, which are typically two weeks or 30 days length.
- Scrum development teams usually attempt to build a **shippable product** increment (properly tested) every sprint.
- The **product backlog** is an ordered list of everything that might be needed in the product.

# Scrum framework

- During sprint planning, the team pulls a small chunk from the top of the product backlog and creates, a **sprint backlog**, and decides how to implement those pieces.
- At the end of the sprint, the work should be potentially shippable: ready to hand to a customer, put on a store shelf, or show to a stakeholder.
- The sprint ends with a **sprint review** and **retrospective**.
- As the next sprint begins, the team chooses another chunk of the product backlog and begins working again.

# Scrum Model: Roles, Meetings and Artifacts

- Roles
  - Product Owner
  - Scrum Team
  - Scrum Master

- Artifacts
  - Product Backlog
  - Product Backlog Item
  - Sprint Backlog
  - Sprint Burn-down Chart

- Meetings
  - Sprint Planning Meeting
  - Daily Scrum
  - Sprint Review
  - Scrum Retrospective Meeting
  - Backlog Refinement Meeting

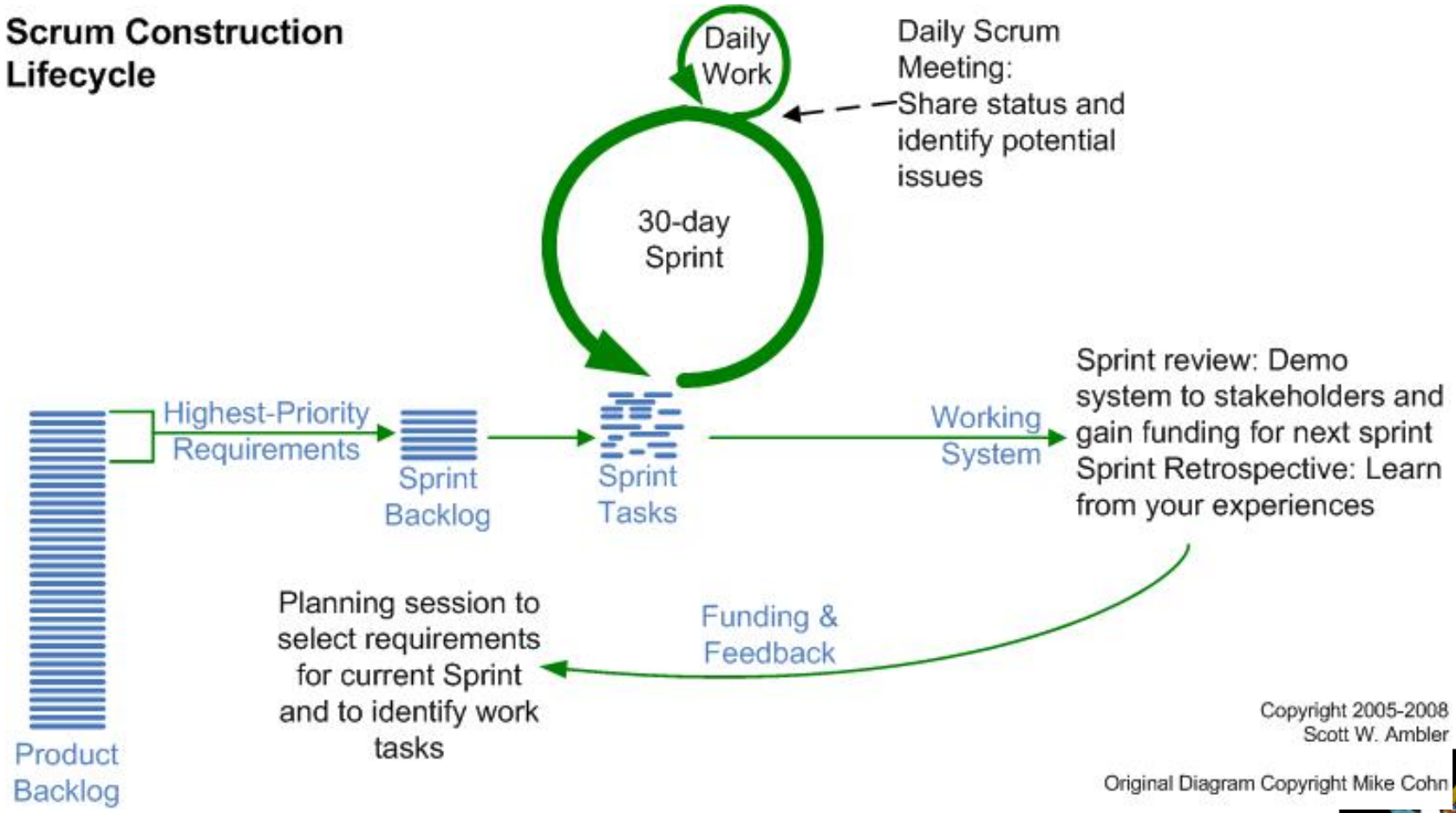
# Scrum framework

- **Scrum Roles**
  - A **Product Owner**
    - Creates a wish list called a **product backlog**.
    - He is responsible for continuously communicating the vision and priorities to the development team.
    - Constantly re-prioritizes the product backlog.
    - Accepts or rejects each product increment.
  - **Scrum Master**
    - **Facilitates** the scrum process.
    - Helps resolve impediments.
    - keeps the team focused on its **goal**.
    - Enforces time boxes.
    - Has no management authority over the team.

# Scrum framework

- The **development team**
  - Cross functional.
  - Self-organizing / self-managing.
  - Intensely collaborative.
  - Has autonomy regarding how to reach commitments.
  - has a certain amount of time — a **sprint** — to complete its work, but it meets each day to assess its progress (**daily Scrum**).

# Agile Practices : Scrum Model



# Scrum Meetings

- **Sprint planning meeting**
  - Happens at the start of each sprint.
  - To negotiate which product backlog items to do during the sprint.
  - Product owner to declare which are the important items.
  - Development team to select the amount of work they feel they can implement during the sprint.
  - Move items from “Product backlog” to “sprint Backlog”.

# Scrum Meetings

- **Daily Scrum meeting**
  - Happens every day at the same time and place.
  - It takes about 15 minutes
  - Each member in the development team summarizes:
    - What he/she did in the previous day.
    - What he/she will do today.
    - What impediments he/she faces.

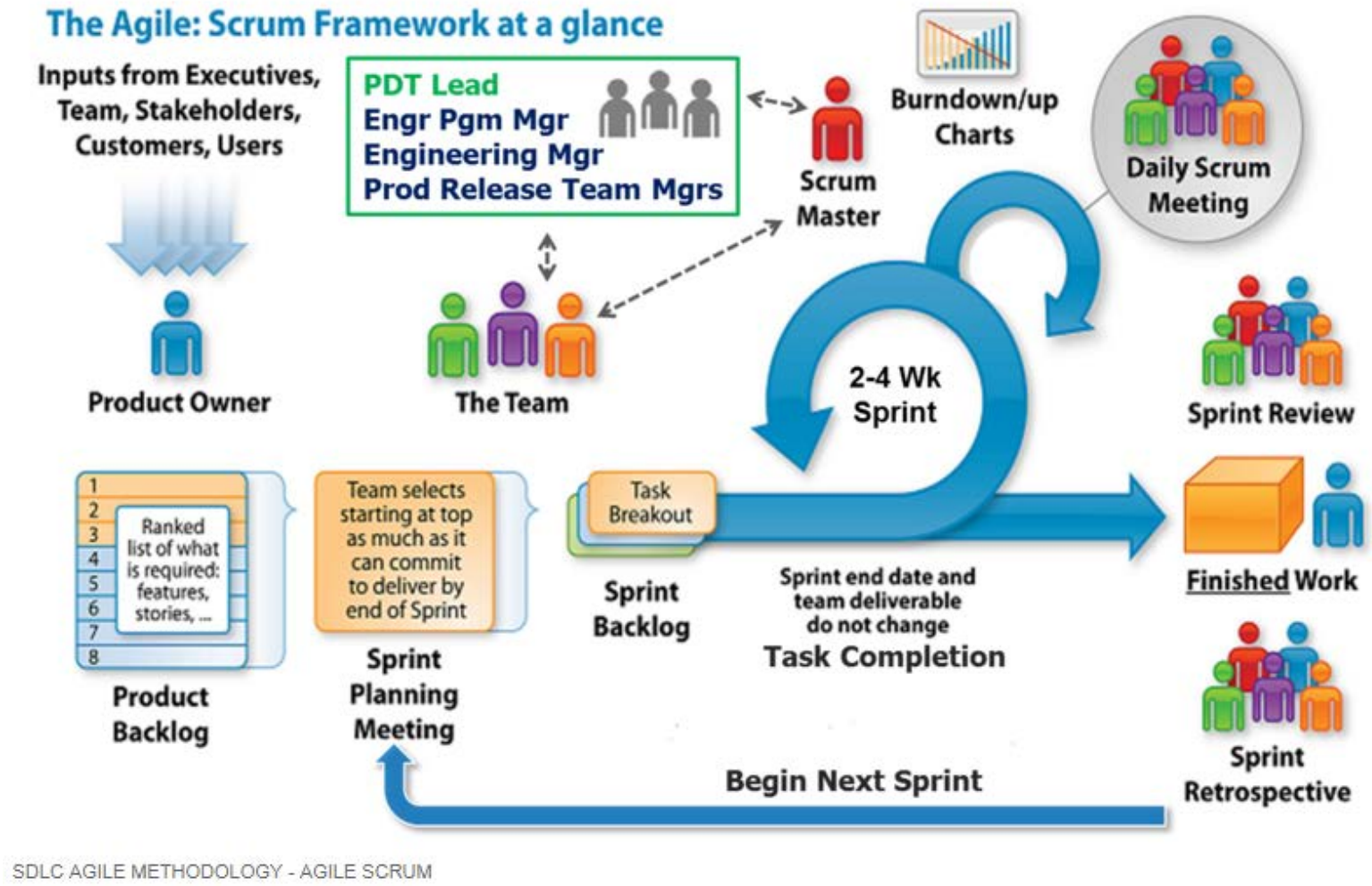
# Scrum Meetings

- **Sprint review meeting**
  - Happens after sprint execution.
  - It features a live demonstration to the product owner and other stakeholders.
  - After demonstration product owner declares which items are considered to be “done”.
  - Unfinished items are returned back to the product backlog.

# Scrum Meetings

- **Sprint Retrospective meeting**
  - At the end of the sprint.
  - Team reflects on its own process.
  - Inspect behaviour and take actions to adapt it for future sprints.

# The Agile: Scrum Framework




https://www.automationlearn.com/home/manual-testing/sdlc-view/

# Scrum Summary

# High Concept Document

- In a nutshell; it addresses “What is the game about?”
- It is a sales tool.
- The high concept document should be 2 to 4 pages long, and take no more than 10 minutes to read.
- The title and your name appear at the top of the first page, and the text begins immediately.
- The document can be outlined along the following sections:
    1. High Concept Statement : “Elevator Pitch”
        - no more than two lines that state the idea of the game

# High Concept Document

2. Features :
   - one page (or less) of bullet points about the features you will want to brag about in your game.
   - Highlight features that are the USPs (Unique Selling Points) of your game.
3. Overview
   - Player motivation : the player's role and goal
   - Genre
   - Storyline
   - Target Audience: Who will buy the game
   - Target Hardware: hardware requirements to run the game
   - Competition : indicate your competitor games and wh[illegible] are better.

# High Concept Document

- Unique selling points: what really stands in your game.
- Design goals: the aims for creating the game
- License: if you intend that the game will be a licensed property.

## 4. Further Information

- Any additional information you think is important to have in your document.

# Summary

- Revision on SDLC
  - Phases of SDLC
- Iterative Process Models
  - Prototyping
  - Spiral
  - Game Development with Spiral Approach
  - Agile Practices
  - High Concept Document

# References


- **“User stories as actives for game development”**

  Victor Schetinger, Cesar Souza, Lisandra Manzoni Fontoura, Cesar Tadeu Pozzer

- http://agilemanifesto.org/